

Claims

I claim:

1. A computer program, called a DSS server, that performs the following tasks:
 - a) Maintains data representing spreadsheets in a persistence layer, such as a filesystem or database;
 - b) Allows users to create, delete, and modify said spreadsheets, including especially the values and derivation formulae of the spreadsheet cells;
 - c) Serves the derived values of said spreadsheet cells to client applications upon valid request;
 - d) Runs calculations when necessary to derive the values of said spreadsheet cells;
 - e) Performs various ancillary functions relating to the above tasks, such as maintaining user information, encrypting data when desired, caching data to speed performance, enforcing an access control policy, and providing an interface for system administration and basic data access.(see Detailed Description section 2).
2. A method of creating a software system that implements some subset of the functionality of the DSS server as described in claim 1) while using data contained in another application as a source for spreadsheet data, by using a generic DSS frontend to communicate with client programs and an adapter built specifically to connect the frontend and the other application (see Detailed Description section 4a).
3. A protocol for controlling communication between DSS servers and DSS client applications, allowing such programs implemented by different vendors to work together seamlessly, which comprises:
 - a) a format for DSS queries and DSS query responses, used for the communication of spreadsheet data, wherein the client program constructs the query and the server replies with the query response;
 - b) a format for DSS commands and DSS command responses, allowing client programs to interact in various ways with the DSS server, such as creating or deleting spreadsheets, adding or editing cell values, adding or deleting users, and other operations;
 - c) techniques to use to ensure the security of these communications.(see Detailed Description section 3).
4. A method for accessing data resident on a DSS server or similar program as stated in claims 1-2), in a client application such as a desktop spreadsheet application, using the protocol described in claim 3) (see Detailed Description section 3a).
5. A method for sending various commands to a DSS server or similar program as described in claims 1-2), to perform operations such as creating or deleting spreadsheets,

adding, editing or removing cells in a spreadsheet, adding or deleting users of the DSS server, and various other operations, from a client application, using the protocol described in claim 3) (see Detailed Description section 3b).

6. A method for creating DSS cell reference formulae in spreadsheet applications to cell data served by computer programs listed in claims 1-2) above, and using the method of data access as described in claim 4) (see Detailed Description section 5).

7. A method of integrating a third party application with a DSS server by modifying the application to report to the DSS server when certain events occur, such as sales, using the protocol described in claim 3), thereby allowing the DSS server to keep track of the reported information in a spreadsheet, and to share that information to DSS client applications (see Detailed Description section 4b).

8. A method for preventing recalculation overload in the DSS server by keeping two distinct sets of spreadsheet data, one public and one private, where the former is updated from the latter at set intervals and otherwise does not change (see Detailed Description section 2d).